Terbit online pada laman web jurnal: **https://jurnal.plb.ac.id/index.php/tematik/index**

TEMATIK

Jurnal **Teknologi Informasi Komunikasi (e-Journal)**

Vol. 12 No. 2 (2025) 200 - 205 ISSN Media Elektronik: 2443-3640

Algoritma Value Based Untuk Pembangunan Bot Trading Yahoo Finance

Value Based Algorithm for Developing Yahoo Finance Trading Bot

Acep Hendra[1], Handoko Supeno[2]

[1]Prodi Sistem Informasi, Fakultas Teknologi dan Informatika, Universitas Informatika dan Bisnis Indonesia

[2]Prodi Teknik Informatika, Fakultas Teknik, Universitas Informatika dan Bisnis Indonesia

[1]acephendra@unibi.ac.id, [2]handoko@unpas.ac.id

Abstract

Advancements in artificial intelligence, particularly reinforcement learning (RL), have driven innovation in automated decision-making within financial markets. Although Deep Reinforcement Learning (DRL) is commonly used, it requires substantial computational resources and lacks transparency. This study proposes the development of a lightweight, transparent, and easily reproducible trading bot based on a value-based RL algorithm (Q-Learning), utilizing open data from Yahoo Finance. The system is built including data acquisition, preprocessing, RL agent design, and strategy evaluation. The Q-Learning agent is trained to determine daily actions (buy, sell, hold) with the objective of maximizing cumulative returns while minimizing risk. Experimental results show that the Q-Learning bot achieves a cumulative return of 180%, a Sharpe Ratio of 1.2, and a win rate of 55%. These findings indicate that tabular Q-Learning has strong potential as an effective, adaptive trading approach with low computational cost.

Keyword: *reinforcement learning, Q-Learning, trading bot, Yahoo Finance, value-based*

Abstrak

Perkembangan kecerdasan buatan, khususnya reinforcement learning (RL), telah mendorong inovasi dalam otomatisasi pengambilan keputusan di pasar keuangan. Meskipun Deep Reinforcement Learning (DRL) sering digunakan, pendekatan ini membutuhkan sumber daya besar dan kurang transparan. Penelitian ini mengusulkan pembangunan bot trading berbasis algoritma value-based RL (Q-Learning) yang ringan, mudah direplikasi, dan menggunakan data terbuka dari Yahoo Finance. Sistem dikembangkan meliputi akuisisi data, preprocessing, desain agen RL, dan pengujian strategi. Agen Q-Learning dilatih untuk menentukan aksi harian (buy, sell, hold) dengan tujuan memaksimalkan cumulative return dan meminimalkan risiko. Hasil eksperimen menunjukkan Q-Learning Bot menghasilkan cumulative return 180%, Sharpe Ratio 1,2, dan win rate 55%. Temuan ini menunjukkan bahwa Tabular Q-Learning memiliki potensi sebagai pendekatan trading adaptif yang efektif dengan biaya komputasi rendah.

Kata kunci: reinforcement learning, Q-Learning, bot trading, Yahoo Finance, value-based.

1. Pendahuluan

Perkembangan teknologi kecerdasan buatan, khususnya di bidang reinforcement learning (RL), telah membawa dampak signifikan terhadap otomatisasi sistem pengambilan keputusan dalam dunia keuangan. Di sektor perdagangan saham, berbagai studi menunjukkan keberhasilan agen RL dalam melakukan asset allocation [1], portfolio optimization[2], hingga aksi buy-sell-hold berdasarkan pembelajaran dari data historis pasar[3]. Namun demikian, mayoritas pendekatan yang digunakan adalah Deep Reinforcement Learning (DRL), yang memerlukan sumber daya komputasi besar, proses pelatihan kompleks, dan terkadang tidak transparan dalam pengambilan keputusan.

Di sisi lain, pendekatan klasik berbasis nilai seperti Q-Learning atau SARSA, yang bersifat lebih ringan dan dapat dijelaskan (interpretable), masih jarang dieksplorasi dalam konteks pembangunan bot trading yang praktis, terutama menggunakan data terbuka seperti yang disediakan oleh Yahoo Finance. Padahal, Yahoo Finance adalah salah satu sumber data paling mudah diakses dan banyak digunakan oleh investor ritel maupun akademisi pemula.

Kondisi ini menimbulkan beberapa tantangan nyata: Tidak adanya sistem bot trading berbasis algoritma value-based sederhana yang dapat secara langsung diterapkan dengan data Yahoo Finance; Minimnya kerangka kerja yang end-to-end, dari proses pengambilan data, pelatihan agen, hingga eksekusi sinyal trading; Kurangnya pendekatan komparatif terhadap strategi konvensional atau sinyal pasar

Diterima Redaksi: 06-11-2025 | Selesai Revisi: 01-12-2025 | Diterbitkan Online: 20-12-2025

sederhana untuk menilai efektivitas agen RL; Keterbatasan literatur lokal maupun open-source tools yang dapat digunakan dalam konteks pendidikan, eksperimen, dan riset skala kecil.

Jika permasalahan ini tidak ditangani, maka: Pemanfaatan RL untuk edukasi dan riset praktis akan tetap terhambat pada pendekatan-pendekatan kompleks dan tidak mudah direplikasi.

Penelitian ini dilakukan untuk bagaimana membangun sebuah bot trading sederhana yang berbasis algoritma value-based reinforcement learning (khususnya Q-Learning), yang mampu mengambil keputusan beli/jual saham secara otomatis, dengan menggunakan data terbuka dari Yahoo Finance, serta dapat diimplementasikan dan diuji secara praktis dalam konteks simulasi pasar.

Kontribusi penelitian ini adalah: Implementasi Q-Learning tabular yg ringan untuk trading saham menggunakan data Yahoo Finance.; Menganalisis performa bot trading yang dihasilkan; Dan mengusulkan algoritma peningkatan yang berdampak pada pelatihan bot trading berbasis tabular q learning.

Dalam beberapa tahun terakhir, reinforcement learning (RL) berkembang pesat dalam algoritma trading, khususnya untuk optimasi portofolio dan pengambilan keputusan otomatis. Gu (2024) [6] mengembangkan Pro Trader RL berbasis behavior cloning untuk meniru pola trader profesional; efektif pada pasar stabil namun kurang fleksibel saat volatilitas tinggi. Studi oleh Yasin & Gill (2024) [7] serta Sun (2024) [8] membahas kerangka umum RL untuk quantitative trading dengan penekanan pada simulasi realistis dan pengujian kondisi pasar, namun kebanyakan menggunakan data global atau premium, berbeda dari penggunaan data terbuka Yahoo Finance pada penelitian ini.

Studi lokal seperti Ridho et al. (2024) [9] dan Sari et al. (2024) [10] menyoroti pentingnya representasi dan normalisasi data dalam penerapan DRL, tetapi belum mengeksplorasi integrasi praktis DRL dengan data Yahoo Finance di konteks Indonesia. Penelitian lain, seperti Varela (2024) [11] dan Μάστορας (2024) [12], membahas implementasi RL pada bot trading nyata, sementara Mohammadshafie et al. (2024) [13] menekankan adaptasi strategi hold/buy/sell berdasarkan profil aset. Pendekatan non-RL seperti permutation decision trees (Ramraj & Nagaraj, 2025) [14] menunjukkan metode klasik masih kompetitif. Praghaadeesh et al. (2024) [15] menggabungkan sentiment analysis dengan RL untuk memperkaya konteks keputusan, dan Liu et al. (2024) [16] menekankan pentingnya dynamic dataset, relevan bagi penggunaan data Yahoo Finance.

2. Metode Penelitian

Penelitian ini bertujuan membangun dan mengevaluasi bot trading saham otomatis berbasis algoritma value-based reinforcement learning (Q-Learning) dengan sumber data terbuka dari Yahoo Finance. Sistem yang dikembangkan terdiri dari tiga komponen utama:

2.1 *Data Acquisition Module*

Proses akuisisi data pada penelitian ini dirancang agar sederhana, dapat direplikasi, dan memanfaatkan sumber data terbuka. Data harga saham historis diperoleh melalui Application Programming Interface (API) Yahoo Finance menggunakan pustaka yfinance pada Python. Modul akuisisi data bertugas mengunduh data harga penutupan harian (adjusted close price) berdasarkan ticker saham yang ditentukan, periode waktu penelitian, dan batas pemisahan data pelatihan serta pengujian. Pertama peneliti membangun fungsi download_data() memanggil API dengan parameter ticker, tanggal awal, dan tanggal akhir. Data yang diterima kemudian diproses untuk memastikan konsistensi struktur, menghapus nilai hilang, dan menyederhanakan kolom menjadi hanya harga penutupan yang sudah disesuaikan. Pendekatan ini memastikan bahwa data yang digunakan telah bebas dari corporate actions seperti stock split atau dividend adjustment, sehingga lebih representatif untuk analisis pergerakan harga.

Selain akuisisi data harga dasar, modul ini juga membangun serangkaian fitur teknikal awal melalui fungsi make_features(). Fitur yang dihasilkan mencakup simple moving average (SMA) periode 5 dan 20 hari, beserta rasio deviasinya terhadap harga terkini, serta momentum 10 hari yang dihitung melalui perubahan persentase harga. Seluruh fitur ini penting untuk menyandikan informasi tren jangka pendek dan menengah yang menjadi masukan bagi agen Q-Learning. Data yang telah dibersihkan dan diperkaya kemudian dipisahkan menjadi dua bagian: data pelatihan (2015–2022) dan data pengujian (2023–2024).

2.2 *Preprocessing*

Tahapan preprocessing dilakukan untuk mengubah data harga mentah menjadi representasi numerik yang dapat diolah oleh agen Reinforcement Learning. Modul ini terdiri dari dua proses utama, yaitu feature discretization dan state encoding, yang bertujuan mengonversi ruang keadaan (state space) menjadi diskrit dan berdimensi terbatas menggunakan metode *quantile-based binning*. Proses ini diimplementasikan melalui fungsi discretize_features() yang memanfaatkan KBinsDiscretizer dari pustaka *scikit-learn*. Setiap fitur dibagi ke dalam enam interval (*bins*) menggunakan strategi *quantile*, sehingga setiap interval mengandung jumlah sampel yang relatif seimbang. Teknik ini dipilih karena mampu mengurangi

DOI: https://doi.org/10.38204/tematik.v12i2.2679

sensitivitas terhadap *outlier* dan menjaga distribusi data tetap stabil. Hasil diskretisasi kemudian diubah ke bentuk bilangan bulat untuk memudahkan pemetaan ke ruang keadaan.

Selanjutnya, modul *preprocessing* menyusun representasi *state* yang menggabungkan informasi posisi portofolio saat ini dan kategori fitur yang telah didiskretisasi. Fungsi state_from_bins() mengodekan seluruh informasi tersebut ke dalam satu indeks keadaan unik. Posisi portofolio yang berada dalam rentang {−1, 0, +1} dimodifikasi menjadi indeks {0, 1, 2}, kemudian digabungkan dengan nilai *bins* melalui teknik *mixed radix encoding*. Proses ini memastikan setiap kombinasi fitur dan posisi memiliki identifikasi unik dalam bentuk indeks integer, sehingga agen Q-Learning dapat memetakan *state-action value* secara efisien. Dengan desain *preprocessing* seperti ini, ruang keadaan menjadi kompak namun informatif, memungkinkan algoritma tabular Q-Learning melakukan pembelajaran secara stabil, tanpa memerlukan jaringan saraf atau arsitektur kompleks seperti pada Deep RL.

3.3 *Reinforcement Learning Agent*

Pada penelitian ini, agen trading dirancang menggunakan algoritma value-based Reinforcement Learning dengan beberapa peningkatan, yaitu Double Q-Learning, multi-step return, dan prioritized updates. Seluruh mekanisme pembelajaran dibangun dalam kerangka tabular sehingga tidak memerlukan jaringan saraf, tetap ringan secara komputasi, namun tetap memiliki kemampuan untuk menangkap dinamika pasar. Agen dikonfigurasi menggunakan sejumlah parameter penting, sebagaimana yang dijabarkan pada Tabel 1.

Tabel 1. Hyperparameter Agen

Hyperparameter	Nilai
N_EPISODES	6000
α (laju pembelajaran)	0.1
γ (faktor diskonto)	0.98
ε max	1.0
ε min	0.05
ε decay	0.01
N STEP	5
PRIORITY REPLAY SIZE	5000
PRIORITY UPDATES	200
PRIORITY EPS	1e-6
PRIORITY DECAY	0.9

Beberapa parameter pelatihan yang penting antara lain jumlah episode pelatihan (N_EPISODES), laju pembelajaran (α = 0.1), faktor diskonto (γ = 0.98), serta strategi eksplorasi ε-greedy dengan nilai awal ε = 1.0 dan penurunan eksponensial hingga batas minimum ε = 0.05. Parameter ini memastikan adanya keseimbangan antara eksplorasi peluang baru dan eksploitasi strategi terbaik yang telah dipelajari. Selain itu, pembelajaran juga memasukkan pearameter trading risiko melalui parameter RISK_AVERSION, biaya transaksi, dan batasan posisi maksimum (MAX_POS), sehingga agen tidak hanya mengejar return tetapi juga memperhatikan stabilitas portofolio sebagaimana yang dijabarkan oleh Tabel 2.

Tabel 2. Trading Parameter

Parameter	Nilai
Trading cost penalty	0.01
RISK AVERSION	0.01
TRANSACTION COST	0.01
INITIAL CASH	100

Untuk mengatasi masalah overestimation bias pada Q-Learning standar, penelitian ini menggunakan pendekatan Double Q-Learning. Dua tabel aksi-keadaan dikonstruksi, yaitu: Q1 dan Q2. Keduanya diperbarui secara bergantian. Ketika memperbarui Q1, pemilihan aksi terbaik dilakukan menggunakan Q1, namun nilai target diambil dari Q2, dan sebaliknya. Strategi ini memberikan estimasi nilai aksi yang lebih stabil dan tidak bias, sehingga meningkatkan ketahanan agen terhadap fluktuasi harga pasar.

Pembelajaran diperkuat dengan mekanisme n-step return, di mana agen tidak hanya mempertimbangkan imbal hasil satu langkah ke depan, tetapi akumulasi imbal hasil selama N_STEP = 5 langkah. Teknik ini memiliki beberapa keuntungan: Mempercepat propagasi informasi nilai dari masa depan ke masa kini; Mengurangi varian estimasi *TD error*; Memperbaiki sensitivitas agen terhadap tren jangka pendek yang relevan dalam trading harian.

Implementasi dilakukan dengan buffer *n-step* yang mencatat rangkaian *state–action–reward* selama beberapa langkah sebelum menghasilkan pembaruan nilai. Kemudian agar pembelajaran lebih efisien, penelitian ini menerapkan prioritized replay, yaitu mekanisme yang memberikan prioritas lebih tinggi kepada pengalaman dengan Temporal-Difference (TD) error besar. Replay buffer disusun sebagai deque dengan kapasitas maksimum 5000 entri, masing-masing berisi informasi state (s), action (a), n-step return (R), state berikutnya (s_next), status terminal (done), nilai prioritas

Proses pelatihan mengintegrasikan tiga teknik peningkatan stabilitas pada algoritma Q-Learning, yaitu Double Q-Learning, multi-step bootstrapping, dan prioritized updates. Ketiga mekanisme ini terbukti memiliki dampak positif pada pelatihan agen sebagai berikut:

Double Q-Learning [4] mengurangi *overestimation bias* yang umum terjadi pada Q-Learning standar ketika aksi terbaik dipilih menggunakan tabel Q yang sama. Dua tabel nilai, Q_1dan Q_2, diperbarui secara bergantian sehingga estimasi aksi dan evaluasi aksi dilakukan oleh tabel yang berbeda.

Multi-step bootstrapping (n-step return) digunakan untuk mempercepat penyebaran sinyal reward ke

DOI: https://doi.org/10.38204/tematik.v12i2.2679

state sebelumnya. Alih-alih menggunakan reward satu langkah, digunakan n-step cumulative return sehingga informasi reward menjadi lebih stabil dan informatif terutama pada pergerakan harga saham.[5]

Prioritized updates digunakan untuk meningkatkan efisiensi belajar. Setiap transisi disimpan dalam prioritized replay buffer dengan bobot prioritas proporsional terhadap besar kesalahan temporal-difference (TD error). Sampel dengan TD error tinggi diproses lebih sering sehingga percepatan konvergensi dapat dicapai sehingga pelatihan agen menjadi lebih optimal.

Gabungan algoritma ketiga mekanisme diatas bekerja pada setiap langkah disetiap episode pelatihan, Dimana pada setiap langkah: Agen memilih aksi dengan *ε-greedy*; Hasil transaksi dihitung berdasarkan perubahan ekuitas, penalti risiko, dan biaya transaksi. Dimana fungsi hadiah didefinisikan pada Persamaan (1) sebagai berikut:

$$R_t = \begin{cases} +1, jika\ profit > 0 \\ -1, jika\ profit < 0 \\ 0, jika\ profit = 0 \end{cases} \quad (1)$$

Transisi dicatat dalam buffer n-step; Setelah horizon n-step terpenuhi, pembaruan Q dilakukan terhadap table Q1. Kemudian setiap beberapa episode konten Q1 akan disalin ke Q2. Persamaan update nilai Q pada tabel didefinisikan pada Persamaan (2) sebagai berikut:

$$Q(s,a) \leftarrow Q(s,a)\ \alpha[r + \gamma max_{a'} Q(s',a') - Q(s,a)] \quad (2)$$

Setelah episode selesai, agen melakukan prioritized replay untuk memperdalam pembelajaran.

Penjelasan komponen-komponen pada Persamaan (2) diatas adalah sebagai berikut: $Q(s,a) \rightarrow$ nilai estimasi quality (keuntungan jangka panjang) saat di state s melakukan aksi a; α adalah learning rate yang digunakan untuk seberapa besar pembaruan memperhitungkan informasi baru (0–1); r adalah $reward$ yang didapat dari transisi $(s,a) \rightarrow s'$; γ adalah factor diskon untuk reward masa depan (0–1)' $\max_{a'} Q(s',a')$ estimasi terbaik untuk aksi berikutnya dari state baru s'; s' adalah state berikutnya; a' adalah aksi berikutnya yang diuji pada a'.

3. Hasil dan Pembahasan

Setelah proses pelatihan selesai, kinerja agen Q-Learning dievaluasi pada periode data uji yang tidak pernah terlihat selama pelatihan. Pada tahap ini, parameter ϵ tidak lagi digunakan untuk eksplorasi; agen bertindak secara deterministik dengan memilih aksi yang memaksimalkan nilai fungsi aksi-keadaan gabungan $Q_{comb}(s,a) = Q_1(s,a) + Q_2(s,a)$. Untuk setiap hari pada periode pengujian, fitur harga saham didiskretisasi dan dikonversi menjadi indeks state menggunakan fungsi state_from_bins(). Berdasarkan state tersebut, agen memilih aksi $a_t \in \{"hold","buy","sell"\}$ dengan $a_t = argmax_a Q_{comb}(s_t,a)$. Aksi yang dipilih kemudian diterjemahkan menjadi perubahan posisi portofolio: 0 → hold, mempertahankan posisi saat ini; 1 → buy, meningkatkan posisi hingga batas maksimum (MAX_POS); 2 → sell, mengurangi posisi hingga batas minimum (misalnya posisi short atau nol).

Setiap keputusan transaksi dieksekusi menggunakan harga penutupan hari berjalan, sehingga menghasilkan pembaruan kas, posisi, dan nilai ekuitas portofolio. Nilai ekuitas harian $Equity_t$ dihitung sebagaimana pada persamaan (3) dibawah ini:

$$Equity_t = Cash_t + Position_t * Price_t \quad (3)$$

Beberapa metrik kuantitatif digunakan untuk mengevaluasi kinerja agen, yaitu:

Cumulative Return: Kinerja total diukur dari selisih ekuitas akhir dengan modal awal sesuai pada Persamaan (4):

$$\text{Total PnL} = Equity_{\text{akhir}} - InitialCash \quad (4)$$

Nilai ini dihitung baik untuk agen Q-Learning maupun strategi buy-and-hold.

Sharpe Ratio: untuk menilai apakah strategi trading menghasilkan keuntungan yang sebanding dengan risiko yang diambil, digunakan Sharpe ratio. Nilai ini dihitung dari perubahan ekuitas harian, sehingga menunjukkan seberapa besar keuntungan yang diperoleh dibandingkan tingkat fluktuasi risikonya.

Win rate menghitung persentase hari di mana perubahan ekuitas harian bernilai positif

Jumlah Transaksi dan Turnover, aktivitas trading dievaluasi melalui: Jumlah transaksi non-nol (frekuensi *buy/sell*); *Turnover*, yaitu total jumlah unit saham yang diperdagangkan sepanjang periode uji.

Metrik ini penting untuk menilai apakah kinerja tinggi dicapai dengan frekuensi transaksi yang wajar dan tidak berlebihan.

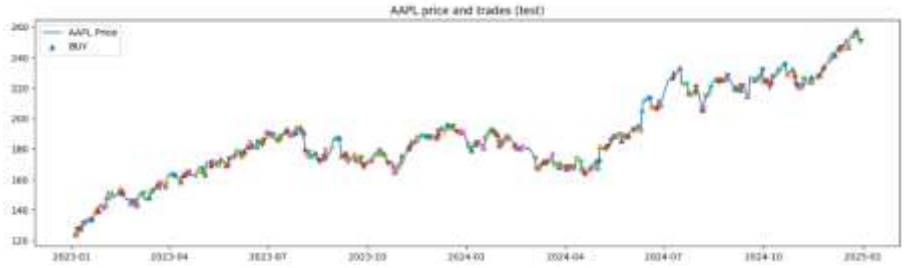
Hasil dari trading bot adalah sebagai berikut: cumulative return 180%; Sharpe Ratio 1.2; win rate 55%, dan Transaction Trade 246.

Untuk memberikan gambaran visual terhadap perilaku agen, dua jenis kurva utama ditampilkan:

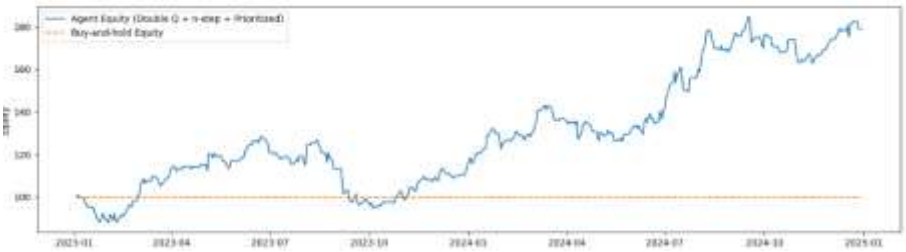
Grafik pertama yang ditunjukkan Gambar 1 menampilkan pergerakan harga saham pada periode uji, disertai penanda aksi beli dan jual yang dilakukan agen. Titik beli ditandai dengan simbol panah ke atas (*marker* ^), sedangkan titik jual dengan panah ke bawah (*marker* v). Visualisasi ini membantu perilaku keputusan agen, disini dapat terihat bahwa agen cenderung mengikuti tren walau masih bersifat terlalu aktif.

DOI: https://doi.org/10.38204/tematik.v12i2.2679

Grafik kedua yang ditunjukkan Gambar 2 menampilkan kurva ekuitas portofolio agen Q-Learning dibandingkan dengan baseline buy-and-hold. Disini terlihat bahwa agen berhasil memberikan keuntungan 80% dari yang awalnya 100 menjadi 180.



Gambar 1. Perilaku Agen



Gambar 2. Performa Agen

Model masih menggunakan quantile discretization sehingga informasi pasar yang kompleks tereduksi menjadi level-level diskrit. Hal ini dapat menyebabkan hilangnya pola halus (fine-grained patterns) pada harga dan volatilitas. Eksperimen hanya dilakukan pada satu aset (misal AAPL) dan tidak mempertimbangkan faktor ekonomi makro, sentimen berita, atau korelasi antar aset yang sering berpengaruh pada keputusan trading di dunia nyata. Biaya transaksi ditetapkan secara tetap, padahal kondisi nyata sangat bervariasi bergantung volatilitas, likuiditas, dan mekanisme pasar.

Meskipun penelitian ini masih menggunakan Tabular Q-Learning, terdapat beberapa arah pengembangan yang dapat meningkatkan performa tanpa perlu beralih ke metode deep learning. Pada sisi representasi state, skema discretization yang saat ini statis dapat diganti dengan adaptive binning yang menyesuaikan lebar bin berdasarkan volatilitas atau distribusi harga, sehingga perubahan pola pasar dapat ditangkap lebih efektif tanpa menambah beban komputasi. Perbaikan lain dapat dilakukan pada mekanisme value update. Penggunaan Q(λ) dengan eligibility traces memungkinkan propagasi reward yang lebih cepat ke state sebelumnya, mempercepat konvergensi sekaligus meningkatkan sensitivitas terhadap rangkaian perubahan harga, tetap dalam kerangka tabular yang ringan. Strategi eksplorasi juga dapat ditingkatkan. Pendekatan ε-greedy yang digunakan saat ini dapat diganti dengan metode yang lebih adaptif seperti Upper Confidence Bound (UCB), yang lebih efisien dalam menyeimbangkan eksplorasi dan eksploitasi, terutama pada lingkungan pasar yang penuh noise dan reward yang jarang. Secara keseluruhan, kombinasi representasi state yang lebih adaptif, update nilai yang lebih efisien, dan eksplorasi yang lebih cerdas berpotensi meningkatkan performa Tabular Q-Learning secara signifikan, tetap dengan kompleksitas rendah dan relevan untuk aplikasi trading nyata.

4. Kesimpulan

Dari eksperimen ini, dapat disimpulkan bahwa pendekatan Tabular Q-Learning memiliki potensi untuk memberikan return yang cukup tinggi dibandingkan strategi pasif. Peningkatan kinerja berasal dari kemampuan Q-Learning ditambah dengan stabilisasi dari double Q learning, horizon pengamatan yang lebih luas dengan multi step, serta konvergensi yang lebih baik dengan prioritized update. Penggabungan beragam Teknik ini memungkinkan agen dapat menyesuaikan posisi berdasarkan pola harga dan indikator teknikal yang di-discretize, sehingga keputusan perdagangan lebih adaptif terhadap kondisi pasar yang berubah-ubah walaupun masih menggunakan teknik tabular yang ringan dan dapat diimplementasikan tanpa memerlukan GPU maupun kebutuhan sumber daya yang besar.

Daftar Rujukan

[1] A. Oshingbesan, E. Ajiboye, P. Kamashazi, and T. Mbaka, "Model-Free Reinforcement Learning for Asset Allocation," ArXiv Prepr. ArXiv220910458, 2022.

[2] J. Jang and N. Seong, "Deep reinforcement learning for stock portfolio optimization by connecting with modern portfolio theory," Expert Syst. Appl., vol. 218, p. 119556, 2023.

[3] S. Otabek and J. Choi, "Multi-level deep Q-networks for Bitcoin trading strategies," Sci. Rep., vol. 14, no. 1, p. 771, 2024.

DOI: https://doi.org/10.38204/tematik.v12i2.2679

[4] Van Hasselt, Hado, Arthur Guez, and David Silver. "Deep reinforcement learning with double q-learning." *Proceedings of the AAAI conference on artificial intelligence*. Vol. 30. No. 1. 2016.

[5] Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., ... & Kavukcuoglu, K. (2016, June). Asynchronous methods for deep reinforcement learning. In *International conference on machine learning* (pp. 1928-1937). PmLR..

[6] Y. H. Gu and others, "Pro Trader RL: Reinforcement learning framework for generating trading knowledge by mimicking the decision-making patterns of professional traders," Expert Syst. Appl., vol. 254, p. 124465, 2024.

[7] A. S. Yasin and P. S. Gill, "Reinforcement Learning Framework for Quantitative Trading," ArXiv Prepr. ArXiv241107585, 2024.

[8] S. Sun, "Reinforcement learning for financial trading: algorithms, evaluations and platforms," 2024.

[9] M. R. Ridho, N. Fajrah, and F. Fifi, "Literatur Review: Penerapan Deep Reinforcement Learning Dalam Business Intelligence," J. Desain Dan Anal. Teknol., vol. 3, no. 2, pp. 96–103, 2024.

[10] E. P. Sari, S. B. Mustamin, M. Atnang, N. Fajar, and others, "Studi Literatur Deep Learning dan Machine Learning untuk Analisis dan Prediksi Pasar Saham: Metodologi, Representasi Data, dan Studi Kasus," J. Teknol. Dan Sains Mod., vol. 1, no. 1, pp. 19–28, 2024.

[11] A. Varela, "Achilles, Neural Network to Predict the Gold Vs US Dollar Integration with Trading Bot for Automatic Trading," ArXiv Prepr. ArXiv241021291, 2024.

[12] Γ. Μστοραζ, "Reinforcement learning-based stock trading: training, evaluation and integration of an agent into a brokerage platform bot".

[13] A. Mohammadshafie, A. Mirzaeinia, H. Jumakhan, and A. Mirzaeinia, "Deep reinforcement learning strategies in finance: Insights into asset holding, trading behavior, and purchase diversity," in World Congress in Computer Science, Computer Engineering & Applied Computing, Springer, 2024, pp. 449–463.

[14] V. Ramraj, N. Nagaraj, and others, "Predicting Stock Prices using Permutation Decision Trees and Strategic Trailing," ArXiv Prepr. ArXiv250412828, 2025.

[15] R. Praghaadeesh, V. Maniappan, S. Doss, and others, "Enhancing Algorithmic Trading Strategies with Sentiment Analysis: A Reinforcement Learning Approach," in 2024 IEEE 3rd World Conference on Applied Intelligence and Computing (AIC), IEEE, 2024, pp. 107–112.

[16] X.-Y. Liu et al., "Dynamic datasets and market environments for financial reinforcement learning," Mach. Learn., vol. 113, no. 5, pp. 2795–2839, 2024.

DOI: https://doi.org/10.38204/tematik.v12i2.2679